DETECTING ERRORS AND IMPUTING MISSING DATA FOR SINGLE LOOP SURVEILLANCE SYSTEMS

Chao Chen*
Department of Electrical Engineering and Computer Science
University of California, Berkeley CA 94720
Tel: (510) 643-5894
chaos@eecs.berkeley.edu

Jaimyoung Kwon
Department of Statistics
University of California, Berkeley CA 94720
Tel: (510) 642-2781, Fax: (510) 642-7892
kwon@stat.berkeley.edu

John Rice
Department of Statistics
University of California, Berkeley CA 94720
Tel: (510) 642-2781, Fax: (510) 642-7892
rice@stat.berkeley.edu

Alexander Skabardonis
Institute of Transportation Studies
University of California, Berkeley CA 94720-1720
Tel: (510) 642-9166, Fax: (510) 642-1246
skabardonis@uclink4.berkeley.edu

Pravin Varaiya
Department of Electrical Engineering and Computer Science
University of California, Berkeley CA 94720
Tel: (510) 642-5270
varaiya@eecs.berkeley.edu

For Presentation and Publication
82nd Annual Meeting
Transportation Research Board
January 2003
Washington, D.C.

November 15, 2002

No WORDS: 4627
Plus 7 Figures (1750)
Plus 4 Tables (1000)
TOTAL: 7377

**Corresponding Author*

ABSTRACT

Single loop detectors provide the most abundant source of traffic data in California, but loop data samples are often missing or invalid. We describe a method that detects bad data samples and imputes missing or bad samples to form a complete grid of 'clean data', in real time. The diagnostics algorithm and the imputation algorithm that implement this method are operational on 14,871 loops in six Districts of the California Department of Transportation.

The diagnostics algorithm detects bad (malfunctioning) single loop detectors from their volume and occupancy measurements. Its novelty is its use of time series of many samples, rather than basing decisions on single samples, as in previous approaches. The imputation algorithm models the relationship between neighboring loops as linear, and uses linear regression to estimate the value of missing or bad samples. This gives a better estimate than previous methods because it uses historical data to learn how pairs of neighboring loops behave. Detection of bad loops and imputation of loop data are important because they allow algorithms that use loop data to perform analysis without requiring them to compensate for missing or incorrect data samples.

INTRODUCTION

Loop detectors are the best source of real time freeway traffic data today. In California, these detectors cover most urban freeways. Loop data provide a powerful means to study and monitor traffic (*2*). But the data contain many holes (missing values) or bad (incorrect) values and require careful 'cleaning' to produce reliable results. Bad or missing samples present problems for any algorithm that uses the data for analysis. Therefore, we need both to detect when data are bad and throw them out, and to 'fill' holes in the data with imputed values. The goal is to produce a complete grid of reliable data. We can trust analyses that use such a complete data set.

We need to detect bad data from the measurements themselves. The problem was studied by the FHWA, Washington DOT, and others. Existing algorithms usually work on the raw 20-second or 30-second data, and produce a diagnosis for each sample. But it's very hard to tell if a single 20-second sample is good or bad unless it's very abnormal. Fortunately, loop detectors don't just give random errors—some loops produce reasonable data all the time, while others produce suspect data all the time. By examining a time series of measurements one can readily distinguish bad behavior from good. Our diagnostics algorithm examines a day's worth of samples together, producing convincing results.

Once bad samples are thrown out, the resulting holes in the data must be filled with imputed values. Imputation using time series analysis has been suggested before, but these imputations are only effective for short periods of missing data; linear interpolation and neighborhood averages are natural imputation methods, but they don't use all the relevant data that are available. Our imputation algorithm estimates values at a detector using data from its neighbors. The algorithm models each pair of neighbors linearly, and fits its parameters on historical data. It is robust, and performs better than other methods.

We first describe the data and types of errors that are observed. We then survey current methods of error detection, which operate on single 20-second samples. Then we present our diagnostic algorithm, and show that it performs better. We then present our imputation algorithm, and show that this method is better than other imputation methods such as linear interpolation.

DESCRIPTION OF DATA

The freeway Performance Measurement System (PeMS) (*1,2*) collects, stores, and analyzes data from thousands of loop detectors in six districts of the California Department of Transportation (Caltrans). The PeMS database currently has 1 terabyte of data online, and collects more than 1GB per day. PeMS uses the data to compute freeway usage and congestion delays, measure and predict travel time, evaluate ramp-metering methods, and validate traffic theories. There are 14,871 main line (ML) loops in the PeMS database from six Caltrans districts. The results presented here are for main line loops. Each loop reports the volume $q(t)$—the number of vehicles crossing the loop detector during a 30-second time interval t, and occupancy $k(t)$—the fraction of this interval during which there is a vehicle above the loop. We call each pair of volume and occupancy observations a sample. The number of total possible samples in one day from ML loops in PeMS is therefore (14871 loops) x (2880 sample per loop per day) = 42 million samples. In reality, however, PeMS never receives all the samples. For example, Los Angeles has a missing sample rate of about 15%. While it's clear when we miss samples, it's harder to tell when a received sample is bad or incorrect. A diagnostics test needs to accept or reject samples based on our assumption of what good and bad samples look like.

EXISTING DATA RELIABILITY TESTS

Loop data error has plagued their effective use for a long time. In 1976, Payne (*3*) identified five types of detector errors and presented several methods to detect them from 20-second and 5-minute volume and occupancy measurements. These methods place thresholds on minimum and maximum flow, density, and speed, and declare a sample to be invalid if they fail any of the tests. Later, Jacobsen and Nihan at the University of Washington defined an 'acceptable region' in the *k*-*q* plane, and declared samples to be good only if they fell inside the region (*4*). We call this the Washington Algorithm. The boundaries of the acceptable region are defined by a set of parameters, which are calibrated from historical data, or derived from traffic theory.

Existing detection algorithms (*3,4,5*) try to catch the errors described in (*3*). For example, 'chattering' and 'pulse break up' cause q to be high, so a threshold on q can catch these errors. But some errors cannot be caught this way, such as a detector stuck in the 'off' (q=0, k=0) position. Payne's algorithm would identify this as a bad point, but good detectors will also report (0,0) when there are no vehicles in the detection period. Eliminating all (0,0) points introduces a positive bias in the data. On the other hand, the Washington Algorithm accepts the (0,0) point, but doing so makes it unable to detect the 'stuck' type of error. A threshold on occupancy is similarly hard to set. An occupancy value of 0.5 for one 30-second period should not indicate an error, but a large number of 30-second samples with occupancies of 0.5, especially during non-rush hours, points to a malfunction.

We implemented the Washington Algorithm in Matlab and tested it on 30-second data from 2 loops in Los Angeles, for one day. The acceptable region is taken from (*4*). The data and their diagnoses are shown in Figure 1. Visually, loop 1 looks good (Figure 1b), and loop 2 looks bad (Figure 1d). Loop 2 looks bad because there are many samples with k=70% and q=0, as well as many samples with occupancies that appear too high, even during non-rush hours, and when loop 1 shows low occupancy. The Washington Algorithm, however, does not make the correct diagnosis. Out of 2875 samples, it declared 1138 samples to be bad for loop 1 and 883 bad for loop 2. In both loops, there were many false alarms. This is because the maximum acceptable slope of q/k was exceeded by many samples in free flow. This suggests that the algorithm is very sensitive to thresholds and needs to be calibrated for California. Calibration is impractical because each loop will need a separate acceptable region, and ground truth would be difficult to get.

There are also false negatives–many samples from loop 2 appear to be bad because they have high occupancies during off peak times, but they were not detected by the Washington Algorithm. This illustrates a difficulty with the threshold method—the acceptable region has to be very large, because there are many possible traffic states within a 30-second period. On the other hand, a lot more information can be gained by looking at how a detector behaves over many sample times. This is why we easily recognize loop 1 to be good and loop 2 to be bad by looking at their $k(t)$ plots, and this is a key insight that led to our diagnostics algorithm.

PROPOSED DETECTOR DIAGNOSTICS ALGORITHM

Design

The algorithm for loop error detection uses the time series of flow and occupancy measurements, instead of making a decision based on an individual sample. It is based on the empirical observation that good and bad detectors behave very differently over time. For example, at any given instant, the flow and occupancy at a detector location can have a wide range of values, and one cannot rule most

of them out; but over a day, most detectors show a similar pattern—flow and occupancy are high in the rush hours and low late at night. Figure 2a and 2b show typical 30-second flow and occupancy measurements. Most loops have outputs that look like this, but some loops behave very differently. Figure 2c and 2d show an example of a bad loop. This loop has zero flow and an occupancy value of 0.7 for several hours during the evening rush hour—clearly, these values must be incorrect. We found 4 types of abnormal time series behavior, and list them in Table 1. Types 1 and 4 are self-explanatory; types 2 and 3 are illustrated in Figure 2c, 2d, and Figure 1b. The errors in Table 1 are not mutually exclusive. For example, a loop with all zero occupancy values exhibits both type 1 and type 4 errors. A loop is declared bad if it is in any of these categories.

We did not find a significant number of loops that have chatter or pulse break up, which would produce abnormally high volumes. Therefore the current form of the detection algorithm does not check for this condition. However, a fifth error type and error check can easily be added to the algorithm to flag loops with consistently high counts.

We developed the Daily Statistics Algorithm (DSA) to recognize error types 1-4 above. The input to the algorithm is the time series of 30-second measurements $q(d,t)$ and $k(d,t)$, where d is the index of the day, and t=0,1,2,...,2879 is the 30-second sample number; the output is the diagnosis $\Delta(d)$ for the dth day: $\Delta(d)$=0 if the loop is good, and $\Delta(d)$=1 if the loop is bad. In contrast to existing algorithms that operate on each sample, the DSA produces one diagnosis for all the samples of a loop on each day.

We use only samples between 5am and 10pm to do the diagnostics, because outside of this period, it's more difficult to tell the difference between good and bad loops. There are 2041 30-second samples in this period, therefore the algorithm is a function of 2041x2=4082 variables. Thus the diagnostic $\Delta(d)$ on day d is a function, $\Delta(d) = f(q(d,a), q(d,a+1),..., q(d,b), k(d,a), k(d,a+1),..., k(d,b))$, where a=5*120=600 is the sample number at 5am, and b=22*120=2640 is the last sample number, at 10pm. To deal with the large number of variables, we first reduce them to four statistics, $S_1,...,S_4$, which are appropriate summaries of the time series. Their definitions are given in Table 2, where $S_j(i,d)$ is the jth statistic computed for the ith loop on the dth day. The decision Δ becomes a function of these four variables. For the ith loop and dth day, the decision whether the loop is bad or good is determined according to the rule

$$\Delta_i(d) = \begin{cases} 1 & \text{if} \quad \begin{array}{l} S_1(i,d) > s_1^* \text{ or} \\ S_2(i,d) > s_2^* \text{ or} \\ S_3(i,d) > s_3^* \text{ or} \\ S_4(i,d) < s_4^* \end{array} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where s_j^* are thresholds on each statistic. These four statistics summarize the daily measurements well because they are good indicators of the four types of loop failure listed in Table 1. This is seen in the histogram of each of statistic displayed in Figure 3. The data are collected from Los Angeles on 4/24/2002. The distribution of each statistic shows two distinct populations. In S_1, for example, there are two peaks at 0 and 2041. This shows that there are two groups of loops—one group of about 4700 loops have very few samples that report zero occupancy, and another group of about 300 that report almost all zeros. The second group is bad, because they have type 1 error. Since all four

distributions are strongly bi-modal, Equation 1 is not very sensitive to the thresholds s_j^* which just have to be able to separate the two peaks in the four histograms in Figure 3. The default thresholds are given in Table 2. The only other parameters of this model are the time ranges, and the definition of S_3, where an occupancy threshold of 0.35 is specified. The DSA uses a total of 7 parameters, listed in Table 3. They work well in all 6 Caltrans districts.

Performance

The DSA algorithm is implemented and run on PeMS data. The last column in Table 1 shows the distribution of the 4 types of errors in District 12 (Orange County) for 31 days in October, 2001. Because we don't have the ground truth of which detectors are actually bad, we must verify the performance of this algorithm visually. Fortunately, this is easy for in most cases, because the time series show distinctly different patterns for good and bad detectors. A visual test was performed on loops in Los Angeles, on data from 8/7/2001. There are 662 loops on Interstate 5 and Interstate 210, out of which 142 (21%) were declared to be bad by the algorithm. We then manually checked the plots of occupancy to verify these results. We found 14 loops that were declared good, but their plots suggested they could be bad. This suggests a false negative rate of 14/(662-142) = 2.7%. There were no false positives. This suggests that the algorithm performs very well.

Real-Time Operation

The detection algorithm described above gives a diagnosis on samples from an entire day. But we are also interested in real-time detection—the validity of each sample as it is received. Therefore what we want is a decision $\hat{\Delta}_i(d,t)$, where d is the current day, and t is the current sample time. We use the simple approximation:

$$\hat{\Delta}_i(d,t) = \Delta_i(d-1) \qquad 2$$

where Δ_i is defined in Equation 1. Equation 2 has two consequences. First, a loop is declared good or bad for an entire day. As a result, we lose some flexibility because we may be throwing away good data from a partially bad loop—this point is discussed in the conclusion section. Second, there is a one-day lag in the diagnosis, which introduces a small error. We estimated the probability of loop failure given the loop status on the previous day, and found Equation 2 to be true for 98% of the time. Therefore, it is a good approximation.

IMPUTATION OF MISSING AND BAD SAMPLES

The Need for Imputation

We model the measurement of each detector as either the actual value or an error value, depending on the status Δ:

$$\begin{aligned} q_{meas,i}(d,t) &= q_{real,i}(d,t)(1-\Delta_i(d)) + \varepsilon_i(d,t)\Delta_i(d), \\ k_{meas,i}(d,t) &= k_{real,i}(d,t)(1-\Delta_i(d)) + \phi_i(d,t)\Delta_i(d), 0 \le t \le 2879 \end{aligned} \qquad 3$$

where $q_{meas,i}$ and $k_{meas,i}$ are the measured values, $q_{real,i}$ and $k_{real,i}$ are the true values, and ε_i and ϕ_i are error values that are independent of $q_{real,i}$ and $k_{real,i}$. We obtained an estimate of the loop status in Equation 2. It tells us to discard the samples from detectors that are declared bad. This leaves holes in the data, in addition to the originally missing samples. This is a common problem–at each sample

time, the user must determine whether it is a good sample or not. An application that analyzes the data must deal with both possibilities.

One approach to missing data is to predict them using time series analysis. Nihan modeled occupancy and flow time series as ARMA processes and predicted values in the near future (*6*); Dailey presented a method of prediction from neighbor loops using a Kalman filter (9). In our case, the errors do not occur randomly, but persist for many hours and days. Time series predictions become invalid very quickly and are inappropriate in such situations. We developed an imputation scheme that uses information from good neighbor loops at only the current sample time. This is a natural way of dealing with missing data and is used by traditional imputation methods. For example, to find the total volume of a freeway location with 4 lanes and only 3 working loops, one may reasonably use the average of the 3 lanes and multiply it by 4. This imputes the missing value using the average of its neighbors. Linear interpolation is another example. Suppose detector i is bad, and is located between detectors j and k which are good. Let x_i, x_j, x_k be their locations, and $x_j < x_i < x_k$, then

$$\hat{q}_i(t) = \frac{(x_i - x_j)q_k(t) + (x_k - x_i)q_j(t)}{x_k - x_j} \qquad 8$$

is the linear interpolation imputation. While these traditional imputation methods are intuitive, they make naive assumptions about the data. Our proposed algorithm, on the other hand, models the behavior of neighbor loops better because it uses historical data.

Linear Model Of Neighbor Detectors

We propose a linear regression algorithm for imputation that models the behavior of neighbor loops using historical data. We find that occupancies and volumes of detectors in nearby locations are highly correlated. Therefore, measurements from one location can be used to estimate quantities at other locations, and a more accurate estimate can be formed if all the neighboring loops are used in the estimation. We define two loops to be *neighbors* if they are in the same location in different lanes, or if they are in adjacent locations. Figure 4 shows a typical neighborhood. We find that both volume and occupancy from neighboring locations are strongly correlated. Figure 5 shows two pairs of neighbors with linearly related flow and occupancies. Figure 6 plots the distribution of the correlation coefficients between all neighbors in Los Angeles. It shows that most neighbor pairs have high correlations in both flow and occupancy.

The high correlation among neighbor loop measurements means that linear regression is a good way to predict one from the other. It is also easy to implement and fast to run. We use the following pair-wise linear model to relate the measurements from neighbor loops:

$$\begin{aligned} q_i(t) &= \alpha_0(i,j) + \alpha_1(i,j)q_j(t) + noise \\ k_i(t) &= \beta_0(i,j) + \beta_1(i,j)k_j(t) + noise \end{aligned} \qquad 9$$

For each pair of neighbors (i,j), the parameters $\alpha_0(i,j)$, $\alpha_1(i,j)$, $\beta_0(i,j)$, $\beta_1(i,j)$ are estimated using five days of historical data. Let $q_i(t)$,$q_j(t)$, t=1,2,...,n be the historical measurements of volume, then

$$\alpha_0(i,j), \alpha_1(i,j) = \arg\max_{\alpha_0', \alpha_1'} \left(\frac{1}{n} \sum_{t=1}^{n} [q_i(t) - \alpha_0' - \alpha_1' q_j(t)]^2 \right) \qquad 10$$

The parameters for density are fitted the same way. We can find parameters for all pairs of loops that report data in our historical database, but some loops never report any data. For them, we use a set of global parameters $\alpha_0^*(\delta,l_1,l_2)$, $\alpha_1^*(\delta,l_1,l_2)$, $\beta_0^*(\delta,l_1,l_2)$, $\beta_1^*(\delta,l_1,l_2)$ that generalize the relationship between pairs of loop in different configurations. For each combination of (relative location, lane of loop 1, lane of loop 2), we have a linear model as follows.

$$\begin{aligned} q_i(t) &= \alpha_0^*(\delta, l_i, l_j) + \alpha_0^*(\delta, l_i, l_j) q_j(t) + noise \\ k_i(t) &= \beta_0^*(\delta, l_i, l_j) + \beta_1^*(\delta, l_i, l_j) k_j(t) + noise \end{aligned} \qquad 11$$

where

$\delta = 0$ if i and j are in the same location on the freeway, 1 otherwise
l_i = lane number of loop i
l_j = lane number of loop j
$l_i, l_j = 1,2,3,...,8$

The global parameters are fitted to data similar to the local parameters. In Los Angeles, there are 60,760 pairs of neighbors (i,j) for 5377 loops; in San Bernardino, there are 3,896 pairs for 466 loops. The four parameters for each pair are computed for these two districts and stored in database tables.

When imputing values for loop i using its neighbors, each neighbor provides an estimate, and the final estimate is taken as the median of the pair wise estimates. Both volume and occupancy imputation are performed the same way. The imputation for volume is

$$\begin{aligned} \hat{q}_{ij}(d,t) &= \alpha_0(i,j) + \alpha_1(i,j) q_j(d,t) \\ \hat{q}_i(d,t) &= \text{median}\{\hat{q}_{ij}(d,t), j : \text{neighbor of } i, \hat{\Delta}_j(d,t) = 0\} \end{aligned} \qquad 12$$

Here $\hat{\Delta}_j(d,t)$ obtained from Equation 2 is the diagnosis of the jth loop—only estimates from good neighbors are used in the imputation. Equation 12 is a way to combine information from multiple neighbors. While this method is suboptimal compared to those with joint probability models, such as multiple regression, it is more robust. Multiple regression models all neighbors jointly, as opposed to the pair-wise model adopted here. Dailey also presented an estimation method based on all neighbors jointly using a Kalman filter (*9*). But we chose the pair-wise model for its robustness--it generates an estimate as long as there is one good neighbor. In contrast, multiple regression needs values at each sample time from all the neighbors. Robustness is also increased by use of the median of $\hat{q}_{ij}$'s instead of the mean, which is affected by outliers and errors in Δ_j.

After one iteration, the imputation algorithm generates estimates for all the bad loops that have at least one good neighbor. We still need to do something for the bad loops that don't have good neighbors. We have not decided on a scheme for how to do this, but there are several alternatives. The current implementation simply iterates the imputation process. After the first iteration, a subset of the bad loops is filled with imputed values—these are the loops with good neighbors. In the second iteration, the set of good loops grows to include those that have been imputed in the previous iteration, so some of the remaining bad loops now have good neighbors. This process continues until all loops are filled, or until all the remaining bad loops don't have any good neighbors. The problem with this method is that the imputation becomes less accurate with each succeeding iteration. Fortunately, most of the bad loops are filled in the first iteration. In District 7 on 4/24/2002, for example, the percentages of filled loops in the first 4 iterations are 90%, 5%, 1%, 1%; the entire grid is filled after 8 iterations. Another alternative is to use the current imputation only for the first n imputations. After that, if there are still loops without values, we can use another method such as

historical mean. In any case, an alternative imputation scheme is required for sample times when there are no good data for any loop.

Performance

We evaluate the performance of this algorithm on data from 4/24/2002. To run this test, we found 189 loops that are themselves good and also had good neighbors. From each loop i, we collected the measured flows and occupancies $q_i(t)$ and $k_i(t)$; we then ran the algorithm to compute the estimated values $\hat{q}_i(t)$ and $\hat{k}_i(t)$ based on neighbors. From these, we found the root mean squared errors for each loop, see Table 4. This table shows that the estimates are unbiased as they should be. The standard deviation of imputation error is small compared to the mean and standard deviation of the measurements. Figure 7 compares the estimated and original values for one loop. They show good agreement.

We also compared the performance of our algorithm against that of linear interpolation. Fifteen triplets of good loops were chosen for this test. Ten of the triplets are loops in the same lane, different locations, while 5 other triplets have their loops in the same location, across 3 lanes. In each triplet, we used two loops to predict the volume and occupancy of the third loop using linear interpolation. In every case, the neighborhood method produced a lower error in occupancy estimates; it produced smaller errors in flow estimates in 10 of 15 locations. Overall, the neighborhood method performed better in the mean and median, as expected.

CONCLUSION

We presented algorithms to detect bad loop detectors from their outputs, and to impute missing data from neighboring good loops. Existing methods of detection evaluate each 20-second sample to determine if it represents a plausible traffic state, but we found that there is much more information in how detectors behave over time. Our algorithm makes diagnoses based on the sequence of measurements from each detector over a whole day. Visually, bad data is much easier to detect when viewed as a time series. We found that our algorithm found almost all of the bad detectors that could be found visually.

Our imputation algorithm estimates the true values at locations with bad or missing data. This is an important functionality, because almost any algorithm that uses the data needs a complete grid of data. Traditionally, the way to deal with missing data is to interpolate from near-by loops. Our algorithm performs better than interpolation because it uses historical information on how the measurements from neighbor detectors are related. We model the volume and occupancy between neighbor loops linearly, and find the linear regression coefficients of each neighbor pair from historical data. This algorithm is simple and robust.

There remain many possibilities for improvements to the algorithms described here. The detection algorithm described here has a time lag. To address this, we are developing a truly real time detection algorithm that incorporates neighbor loop measurements as well as the past day's statistics. While the linear model describes most neighbor pairs, some pairs have non-linear relationships, so a more general model may be better. Another area of improvement is the handling of entire blocks of missing data. The current imputation algorithm needs a large number of good loops to impute the rest, but it doesn't work if most or all the loops are bad for a sample time. We need a method to handle this situation.

Single loop data diagnostics is an important area of research. While loop detectors are the most abundant source of traffic information, the data are sometimes bad or missing. The algorithms we presented construct a complete grid of clean data in real time. They simplify the design of upper level algorithms and improve the accuracy of analysis based on loop data.

ACKNOWLEDGEMENTS

This study is part of the PeMS project, which is supported by grants from Caltrans to the California PATH Program. We are very grateful to engineers from Caltrans Districts 3, 4, 7, 8 and 12 and Headquarters for their encouragement, understanding, and patience. They continue to shape the evolution of the PeMS vision, to monitor its progress, to champion PeMS within Caltrans. Without their generous support this project would not have reached such a mature stage.

The contents of this paper reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of or policy of the California Department of Transportation. This paper does not constitute a standard, specification or regulation.

REFERENCES

1. http://transacct.eecs.berkeley.edu

2. Chen, C., K. Petty, A. Skabardonis, and P. Varaiya. Freeway Performance Measurement System: Mining Loop Detector Data. *Transportation Research Record No.1748*, Transportation Research Board, Washington, D.C., 2001, pp 96-102

3. Payne, H.J., E.D. Helfenbein and H.C. Knobel. *Development and testing of incident detection algorithms*, FHWA-RD-76-20, Federal Highway Administration, Washington DC 1976.

4. Jacobson, L., Nihan, N., and J. Bender, Detecting Erroneous Loop Detector Data in a Freeway Traffic Management System. *Transportation Research Record* 1287, Transportation Research Board, Washington, D.C., 1990, pp 151-166

5. Cleghorn, D., F. Hall, and D. Garbuio. Improved Data Screening Techniques for Freeway Traffic Management Systems. *Transportation Research Record* 1320, Transportation Research Board, Washington, D.C., 1991, pp 17-31.

6. Nihan, N. (1997) Aid to Determining Freeway Metering Rates and Detecting Loop Errors. *Journal of Transportation Engineering*, Vol 123, No 6, November/December 1997, pp 454-458.

7. Turochy, R.E. and B.L. Smith. A New Procedure for Detector Data Screening in Traffic Management Systems. *Transportation Research Record* 1727, Transportation Research Board, Washington, D.C., 2000, pp 127-131

8. http://transacct.eecs.berkeley.edu/~chaos/Projects/DataQuality/Timeseries/VdsStats2.html.

9. Dailey, D.J. *Improved error detection for inductive loop sensors***,** WA-RD 3001 Washington State Dept. of Transportation, May 1993.

10. Davis, G. and N. Nihan. Using Time-Series Designs to Estimate Changes in Freeway Level of Service, Despite Missing Data. *Transportation Research. Part A,* Vol 18A, no. 5/6, Oct/Dec 1984, pp. 431-438.

LIST OF TABLES

LIST OF FIGURES

Table 1 Error Types.

Error Type	Description	Likely Cause	Fraction of loops in District 12
1	Occupancy and flow are mostly zero	Stuck off	5.6%
2	Non-zero occupancy and zero flow, see Figure 2c and 2d.	Hanging on	5.5%
3	Very high occupancy, see Figure 1d	Hanging on	9.6%
4	Constant occupancy and flow	Stuck on or off	11.2%
All Errors			16%

Table 2 Statistics for diagnostics.

Name	Definition	Description
S_1(i,d)	$\sum_{a \le t \le b} 1(k_i(d,t) = 0)$	number of samples that have occupancy = 0.
S_2(i,d)	$\sum_{a \le t \le b} 1(k_i(d,t) > 0) 1(q_i(d,t) = 0)$	number of samples that have occupancy>0 and flow=0
S_3(i,d:	$\sum_{a \le t \le b} 1(k_i(d,t) > k^*), k^* = 0.35$	number of samples that have occupancy > k^* (=0.35)
S_4(i,d)	$(-1) \sum_{x: p(x)>0} \hat{p}(x) \log(\hat{p}(x)),$ $\hat{p}(x) = \sum_{a \le t \le b} 1(k_i(d,t) = x) \Big/ \sum_{a \le t \le b} 1$	*entropy* of occupancy samples – a well-known measure of the "randomness" of a random variable. If $k_i(d,t)$ is constant in t, for example, its entropy is zero.

Table 3 Parameters of the Daily Statistics Algorithm, and their default settings

Parameter	Value
k^*	0.35
s_1^*	1200
s_2^*	50
s_3^*	200
s_4^*	4
a	5am
b	10pm

Table 4 Performance of imputation.

Quantity	Mean	Standard Deviation	Mean Absolute Error	Standard Deviation of Error	Mean Error
Occupancy	0.085	0.061	0.013	0.021	0.001
Volume (vph)	1220	527	132	201	6

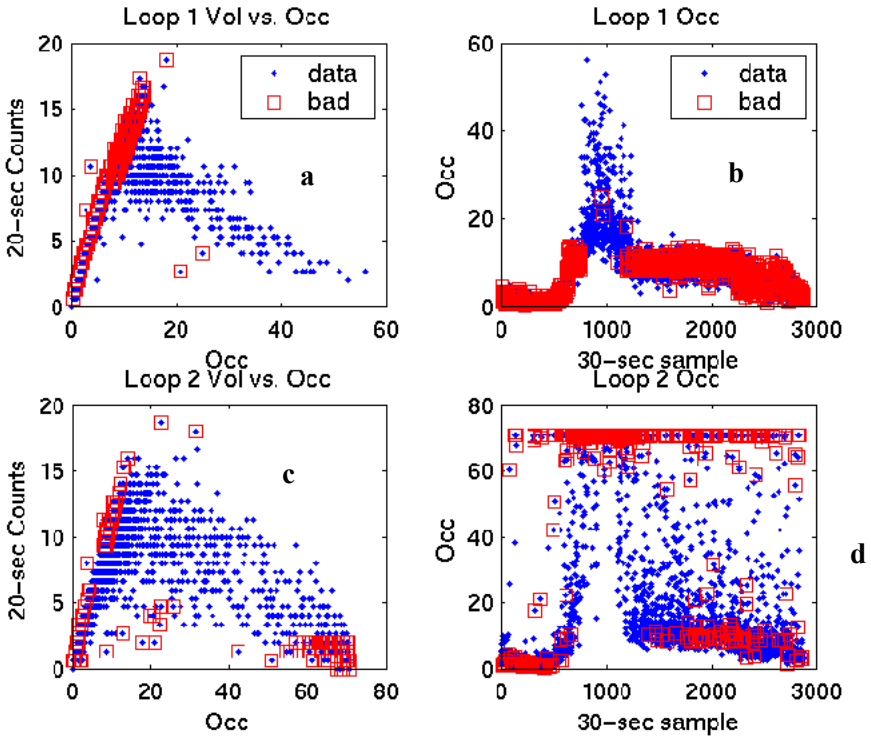

Figure 1 The Washington Algorithm on two loops. Loop 1 and 2 are in Los Angeles, I-5 North, postmile 7.8, lanes 1 and 2; data collected on 8/7/2001. Occupancy is in percent.

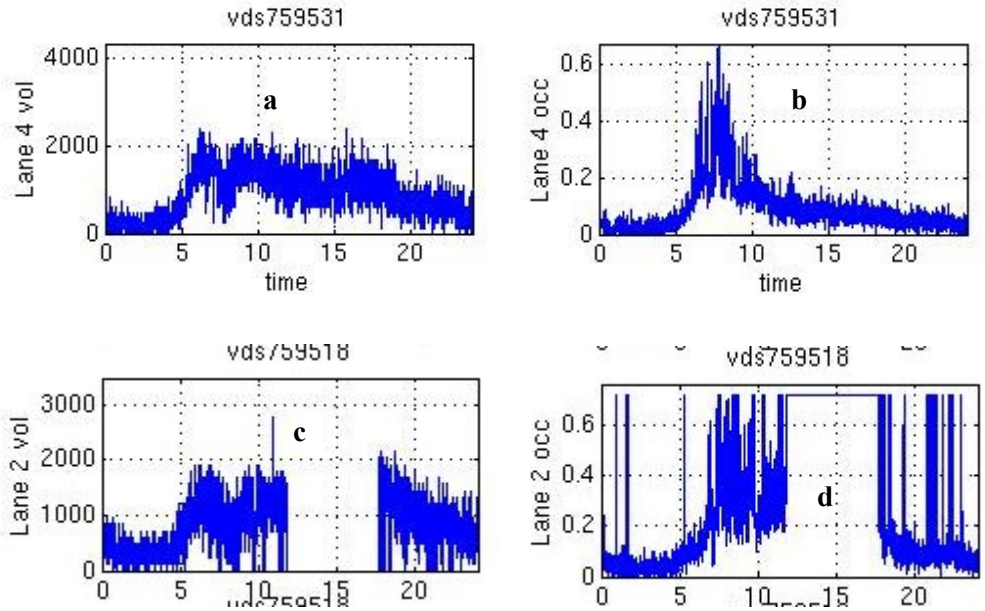

Figure 2 Typical and abnormal 30-sec flow (left) and occupancy measurements.

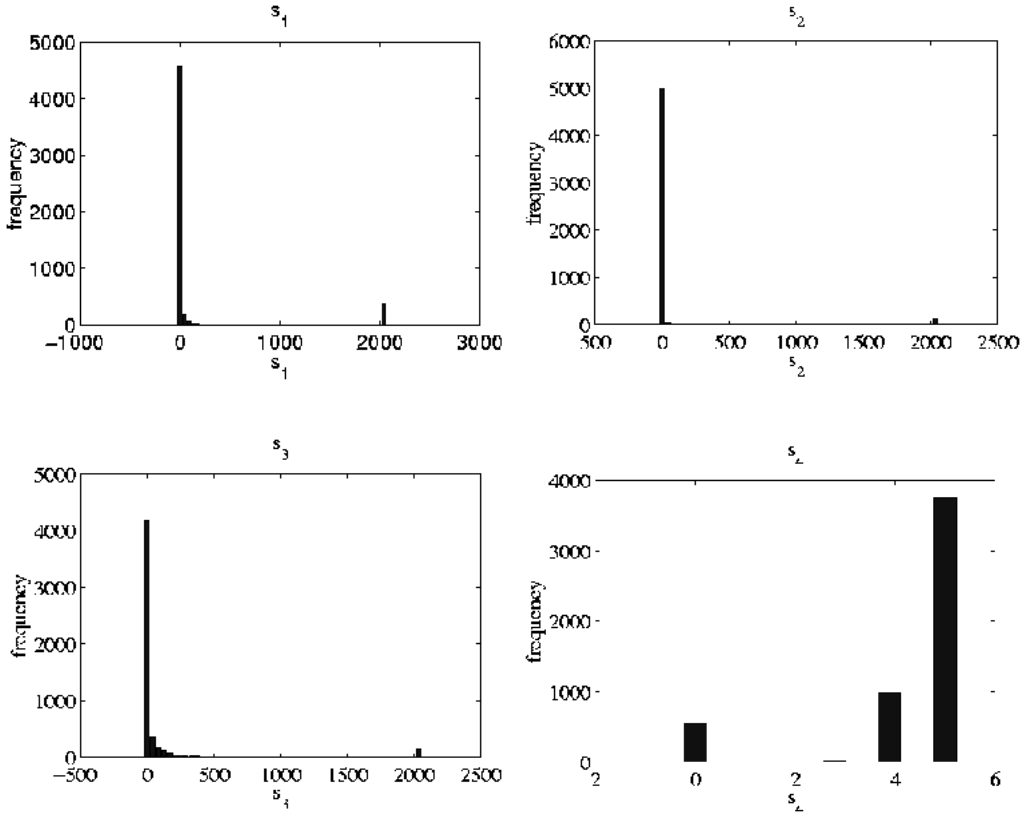

Figure 3 Histograms of S_1 - S_4.

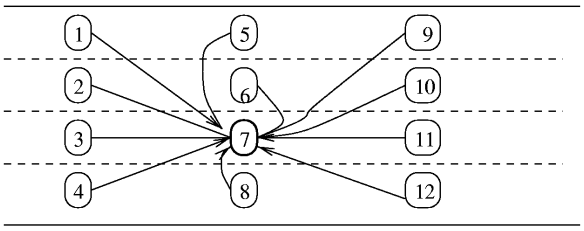

Figure 4 Example of neighboring loops.

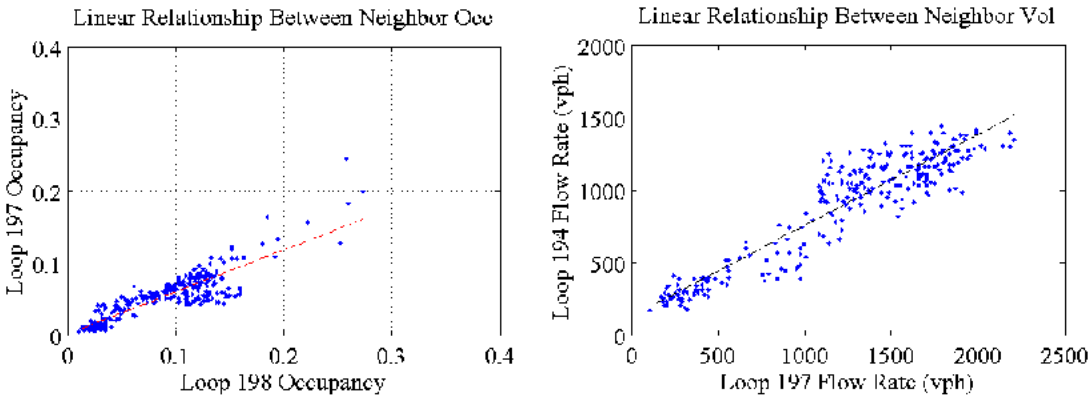

Figure 5 Scatter plot of occupancies and flows from two pairs of neighbors.

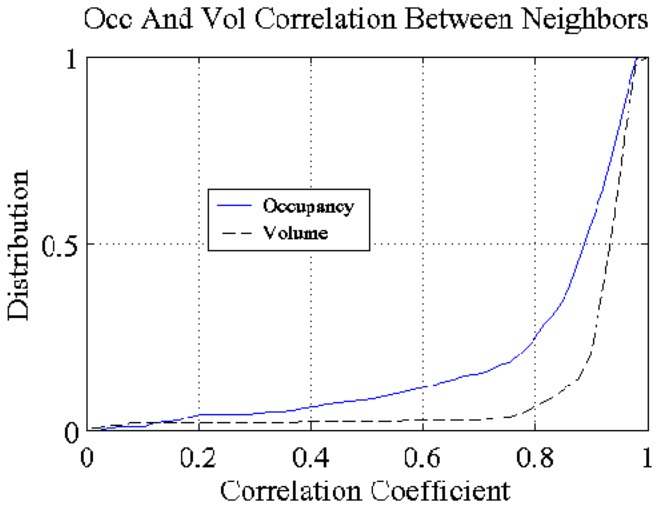

Figure 6 Cumulative distribution of the correlation coefficients between neighbors.

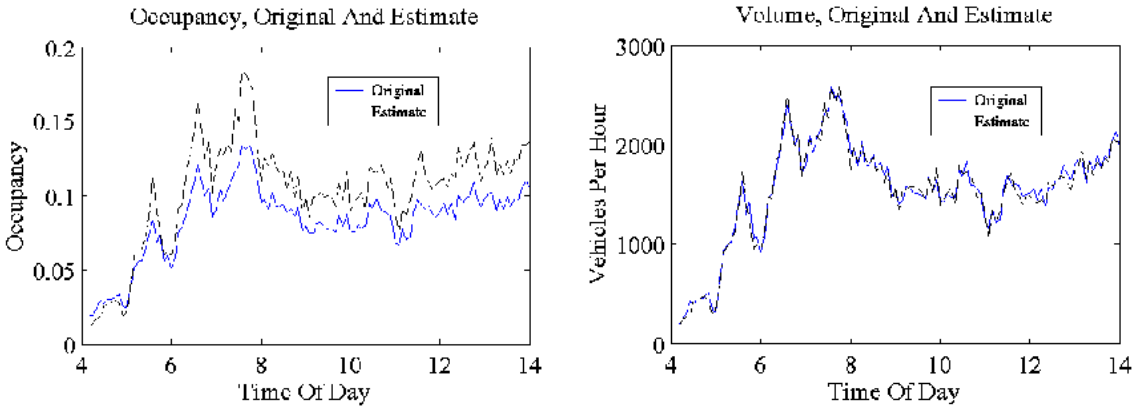

Figure 7 Original and estimated occupancies and flows for a good loop.